

# 遠隔制御に適した圧縮センシング手法の検討

永原 正章<sup>†</sup>

<sup>†</sup> 京都大学情報学研究科 〒606-8501 京都市左京区吉田本町

**あらまし** 無線通信路のように帯域制限された通信路を介した遠隔制御系では、制御信号をいかに効果的に表現し送信するかが重要な問題となる。この問題に対して、圧縮センシングの枠組みでの新しい手法が提案され、 $\ell^1$ - $\ell^2$  最適化による送信信号のスパース表現が有効であることが示された。しかし、圧縮センシングにおいては、 $\ell^1$ - $\ell^2$  最適化のみならず、様々な手法が存在する。本研究では、数値シミュレーションを用いて、遠隔制御系に適した手法を比較検討し、その優劣を考察する。

**キーワード** 遠隔制御、圧縮センシング、スパース表現

## Comparative Study of Compressed-Sensing Methods for Remote Control

Masaaki NAGAHARA<sup>†</sup>

<sup>†</sup> Graduate School of Informatics, Kyoto University Sakyo-ku, Yoshida-Honmachi, Kyoto, 606-8501 Japan

**Abstract** In remote control systems, efficient representation of control signals is one of the crucial issues because of bandwidth-limitedness of the communication channel, such as a wireless communication link, between the controller and the controlled object. Recently, a new method based on compressed sensing has been proposed, in which control signals are sparsely representation based on  $\ell^1$ - $\ell^2$  optimization. There exist however so many methods other than  $\ell^1$ - $\ell^2$  optimization for compressed sensing. In this study, we perform a comparative study of sparsity-promoting methods in compressed sensing, and reveal their advantages and disadvantages by simulation in view of remote control over rate-limited networks.

**Key words** remote control, compressed sensing, sparse representation

### 1. はじめに

遠隔制御系では、制御対象とコントローラが物理的に離れており、センサからの観測信号やアクチュエータへの制御信号は、無線伝送路など伝送帯域の制約が強い通信路を介してやりとりされる。特にデジタル通信を用いた系では、通信路における雑音等の原因によりビット誤りが生じる可能性があり、それを避けるために、誤り訂正符号等で冗長化した信号を送る必要がある。このとき、送信信号に対する伝送帯域の制約はより強いものとなり、したがって、与えられた伝送帯域で効果的に制御信号を送るための手法が重要である。

この問題に対し、最近、筆者らによって、圧縮センシングの手法を用いた制御入力信号の情報圧縮法を提案した [5], [6]。ここで、圧縮センシングとは、元のデータがスパースであるという仮定を置くことによって、非常に少ない観測データから元のデータをかなり正確に復元できる手法であり、近年、とくに信号処理や情報理論の分野で盛んに研究されている [3], [9]。この手法の特徴は、スパースさの指標であるベクトルの非ゼロ要素の数（しばしば“ $\ell^0$  ノルム”と呼ばれる）の最小化問題（これは

組み合わせ最適化であり、問題のサイズに対し、計算量が指数的に増大する）を、欲張り法 [4], [7], [8] や凸緩和 [1], [2], [11] などの手法によって効率よく解くことにある。文献 [5], [6] では、問題を  $\ell^1$ - $\ell^2$  最適化問題として定式化し、その高速解法の一つである繰り返し縮小法 (Fast Iterative Shrinkage-Thresholding Algorithm; FISTA) により、遠隔制御系における制御信号の効率的なスパース表現の手法を提案している。本稿では、遠隔制御の問題に対し、OMP (Orthogonal Matching Pursuit) [4], [8] や CoSaMP (Compressive Sampling Matching Pursuit) [7] などの欲張り法や、上述の FISTA および NESTA (Nesterov's Algorithm) [2] による凸緩和法を導入し、数値シミュレーションを用いて、遠隔制御系に適した手法を比較検討し、その優劣を考察する。

### 記 法

ベクトル  $\mathbf{v} = [v_1, \dots, v_n]^\top \in \mathbb{R}^n$  に対して、その  $\ell^1$  ノルムおよび  $\ell^2$  ノルムを

$$\|\mathbf{v}\|_1 := \sum_{i=1}^n |v_i|, \quad \|\mathbf{v}\|_2 := \left( \sum_{i=1}^n v_i^2 \right)^{1/2}$$

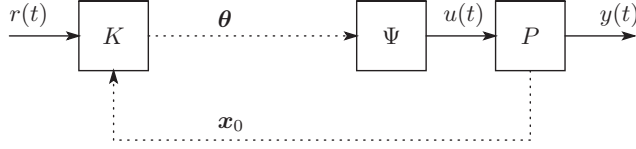


図 1 遠隔制御系：破線は通信路を表す。

で定義する。また、ベクトル  $\mathbf{v}$  の非ゼロ要素のインデックスの集合を

$$\text{supp}(\mathbf{v}) := \{j : v_j \neq 0\}$$

で表し、ベクトル  $\mathbf{v}$  の  $\ell^0$  ノルムを

$$\|\mathbf{v}\|_0 := |\text{supp}(\mathbf{v})|$$

で定義する（ただし、 $|\text{supp}(\mathbf{v})|$  は集合  $\text{supp}(\mathbf{v})$  の要素数を表す）。行列  $\Phi$  に対して、そのノルムを

$$\|\Phi\| := \max_{\|\mathbf{v}\|_2 \neq 0} \frac{\|\Phi \mathbf{v}\|_2}{\|\mathbf{v}\|_2} = \sigma_{\max}(\Phi)$$

で表す。ここで  $\sigma_{\max}(\Phi)$  は行列  $\Phi$  の最大特異値である。実数  $x \in \mathbb{R}$  に対して

$$\text{sgn}(x) := \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0, \end{cases} \quad (x)_+ := \max\{x, 0\}$$

と定義する。区間  $[0, T]$  ( $T > 0$ ) 上の実数値 Lebesgue 2 乗可積分関数の全体を  $L^2[0, T]$  と書く。  $f, g \in L^2[0, T]$  に対して、内積を

$$\langle f, g \rangle := \int_0^T f(t)g(t)dt$$

で表し、ノルムを  $\|f\|_2 := \langle f, f \rangle^{1/2}$  で定義する。

## 2. 問題設定

次の連続時間線形時不変システム  $P$  を考える：

$$P : \begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{b}u(t), \\ y(t) = \mathbf{c}^\top \mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^\nu, \quad t \in [0, T], \end{cases} \quad (1)$$

ここで  $A \in \mathbb{R}^{\nu \times \nu}$ ,  $\mathbf{b}, \mathbf{c} \in \mathbb{R}^{\nu \times 1}$  とする。また、この系は漸近安定、すなわち行列  $A$  のすべての固有値の実部は負であると仮定する。この制御対象に対し、次の追従問題を考える。すなわち、時刻  $t = 0$  から  $t = T$  ( $T > 0$ ) までの参照信号  $\{r(t)\}_{t \in [0, T]}$  が与えられたとき、

$$E(u) := \|y - r\|_2^2 = \|Pu - r\|_2^2 \quad (2)$$

を小さくするような  $u \in L^2[0, T]$  を設計したい。

さらに、この制御系は、図 1 に示すような遠隔制御系であると仮定する。すなわち、コントローラ  $K$  と制御対象  $P$  とは、通信路を介して情報がやり取りされる。ここで、制御信号  $u$  は次で定義される  $L^2[0, T]$  の部分空間  $V_M$  の元とする：

$$V_M := \text{span}\{\psi_m : m = -M, \dots, M\}.$$

ただし、 $\{\psi_m\}$  は次で定義される Fourier 基底である：

$$\psi_m := \frac{1}{\sqrt{T}} \exp(j\omega_m t), \quad \omega_m := \frac{2\pi m}{T}.$$

図 1 において、 $\Psi$  は Fourier 展開係数ベクトル  $\boldsymbol{\theta} \in \mathbb{R}^{2M+1}$  から、次式を用いて制御入力信号  $u \in V_M$  に変換する：

$$u(t) = \sum_{i=1}^N \theta_m \psi_m(t), \quad t \in [0, T]. \quad (3)$$

さらに、参照信号  $r$  も部分空間  $V_M$  の元、すなわち  $r \in V_M$  と仮定する。

次に (2) で定義される評価関数  $E(u)$  を離散化するために、まず、周波数  $2\omega_M$  で信号  $y$  と  $r$  をサンプリングする。すなわち、 $t_n := T(n-1)/(N-1)$ ,  $n = 1, 2, \dots, 2M+1 =: N$  を用いて、

$$E_d(u) = \frac{T}{N-1} \sum_{n=1}^N |y(t_n) - r(t_n)|^2$$

と離散化する。このとき、次式が成り立つ [6]：

$$E_d \left( \sum_{m=-M}^M \theta_m \psi_m \right) = \frac{T}{N-1} \|\mathbf{G}\boldsymbol{\theta} + \mathbf{H}\mathbf{x}_0 - \mathbf{r}\|_2^2.$$

ここで、

$$\mathbf{G} := \begin{bmatrix} \langle \phi_1, \psi_{-M} \rangle & \dots & \langle \phi_1, \psi_M \rangle \\ \langle \phi_2, \psi_{-M} \rangle & \dots & \langle \phi_2, \psi_M \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi_N, \psi_{-M} \rangle & \dots & \langle \phi_N, \psi_M \rangle \end{bmatrix} \in \mathbb{C}^{N \times N}, \quad (4)$$

$$\phi_n(t) := \begin{cases} \mathbf{c}^\top \exp[(t_n - t)A] \mathbf{b}, & \text{if } 0 \leq t < t_n \leq T, \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathbf{r} := \begin{bmatrix} r(t_1) \\ r(t_2) \\ \vdots \\ r(t_N) \end{bmatrix} \in \mathbb{R}^N, \quad \mathbf{H} := \begin{bmatrix} \mathbf{c}^\top \exp(t_1 A) \\ \mathbf{c}^\top \exp(t_2 A) \\ \vdots \\ \mathbf{c}^\top \exp(t_N A) \end{bmatrix} \in \mathbb{R}^{N \times \nu}.$$

である。

今考えている遠隔制御系では、ベクトル  $\boldsymbol{\theta}$  が通信路を介して送信される。ここで、通信路の容量に制約があるのが通常で、 $\boldsymbol{\theta}$  のデータ量は少ない方が望ましい。そこで、圧縮センシングの考え方 [3], [9] に従い、送信信号  $\boldsymbol{\theta}$  のスパース表現を考える。

まず、次のランダム行列  $U$  を定義する：

$$U = \begin{bmatrix} \mathbf{e}_{i(1)} \\ \mathbf{e}_{i(2)} \\ \vdots \\ \mathbf{e}_{i(K)} \end{bmatrix} \in \{0, 1\}^{K \times N},$$

ここで  $i(1) < i(2) < \dots < i(K)$  は  $\{1, 2, \dots, N\}$  上の一様分布から生成される確率変数であり、

$$\mathbf{e}_i := [0, \dots, 0, \overset{i}{1}, 0, \dots, 0], \quad i = 1, 2, \dots, N$$

である。このランダム行列  $U$  を用いて、次の最適化によりス

---

**Algorithm 1** OMP for sparse control vector  $\theta$ 

---

**Require:**  $\alpha \in \mathbb{R}^K$  {observed vector}**Ensure:**  $\theta$  {sparse control vector} $\theta[0] := \mathbf{0}, \mathbf{r}[0] := \alpha - \Phi\theta[0], \mathcal{S}[0] := \text{supp}\{\theta[0]\} = \emptyset.$  $k := 0.$ **while**  $\|\mathbf{r}[k]\|_2 > \epsilon$  **do**  **for**  $j = 1$  **to**  $N$  **do**

$$z_j := \frac{\phi_j^\top \mathbf{r}[k]}{\|\phi_j\|_2^2} = \arg \min_{z \in \mathbb{R}} \|\phi_j z - \mathbf{r}[k]\|_2^2.$$

$$e_j := \|\phi_j z_j - \mathbf{r}[k]\|_2^2.$$

**end for**  Find a minimizer  $j_0 \notin \mathcal{S}[k]$  such that  $e_{j_0} \leq e_j$ , for all  $j \notin \mathcal{S}[k]$ .   $\mathcal{S}[k+1] := \mathcal{S}[k] \cup \{j_0\}$ 

$$\theta[k+1] := \arg \min_{\text{supp}\{\theta\}=\mathcal{S}[k+1]} \|\Phi\theta - \alpha\|_2^2.$$

$$\mathbf{r}[k+1] := \alpha - \Phi\theta[k+1].$$

 $k := k+1.$ **end while****return**  $\theta = \theta[k].$ 

---

パースなベクトル  $\theta$  を求める：

$$(P0) \quad \min_{\theta} \|\theta\|_0 \text{ subject to } \|\Phi\theta - \alpha\|_2 \leq \epsilon.$$

ここで、 $\Phi := UG$ ,  $\alpha = U(\mathbf{r} - H\mathbf{x}_0)$  である。この最適化は、組み合わせ最適化であり、ベクトル  $\theta$  のサイズが大きくなると、計算量が指数関数的に増大する。特に、本稿で考察するような遠隔制御系では、最適化計算による遅れが制御系の安定性や性能に大きく影響するため、計算量はなるべく小さくしたい。この問題に対して、欲張り法や凸緩和などにもとづく高速解法が多く存在する。次節では、それらのうちいくつかを紹介する。

### 3. 最適化手法

(P0) の最適化問題を解く方法はさまざまにあり、主要な計算方法として、欲張り法や凸緩和法、Bayesian アプローチ、非凸最適化、総当り法などがある [10]。この中で、本稿では特に欲張り法および凸緩和を採用し、それらの遠隔制御系における優劣を比較する。

#### 3.1 欲張り法

欲張り法のアプローチのうち、Orthogonal Matching Pursuit (OMP) と呼ばれる手法は、圧縮センシングの最も初期の段階で提案されており [4], [8]、現在でも多くの場合に有効な手法として知られている。(P0) を解くための OMP のアルゴリズムを **Algorithm1** に示す。また、最適ベクトル  $\theta$  のスパースさ、すなわち、 $\|\theta\|_0$  が既知の場合は、圧縮センシングにおける制限等長性 (Restricted Isometry Property, RIP) を巧みに使って OMP を高速化させた CoSaMP (Compressive Sampling Matching Pursuit) [7] と呼ばれるものも存在する。そのアルゴリズムは、参考文献 [7] を参照していただきたい。

#### 3.2 凸緩和法

(P0) の最適化を解く別の方法として凸緩和法がある。凸緩和法とは、(P0) の  $\ell^0$  ノルムの項を  $\ell^1$  ノルムで置き換える方法である。すなわち、問題 (P0) は次の凸最適化 (P1) で近似

---

**Algorithm 2** NESTA for sparse control vector  $\theta$ 

---

**Require:**  $\alpha \in \mathbb{R}^K$  {observed vector}**Ensure:**  $\theta$  {sparse control vector} $\theta[0] := \mathbf{0}, \mathbf{w}[0] := \text{sat}_\delta(\theta[0]), \mathbf{v}[0] := \frac{1}{2}\mathbf{w}[0].$  $k := 1.$ **repeat**

$$\mathbf{q}[k] := \theta[k] - \delta\mathbf{w}[k].$$

$$\lambda_k := \max\left\{0, \frac{1}{\delta\epsilon} \|\alpha - \Phi\mathbf{q}[k]\|_2 - \frac{1}{\delta}\right\}.$$

Solve the following linear equation for  $\eta[k]$ :

$$(I + \delta\lambda_k\Phi^\top\Phi)\eta[k] = \delta\lambda_k\Phi^\top\alpha + \mathbf{q}[k].$$

Solve the following linear equation for  $\zeta[k]$ :

$$(I + \delta\lambda_k\Phi^\top\Phi)\zeta[k] = \delta\lambda_k\Phi^\top\alpha + \theta[0] + \mathbf{q}[k].$$

$$\theta[k+1] := \frac{2}{k+4}\zeta[k] + \frac{k+2}{k+4}\eta[k].$$

$$\mathbf{w}[k+1] := \text{sat}_\delta(\theta[k+1]).$$

$$\mathbf{v}[k+1] := \frac{1}{2k+4}\mathbf{w}[k+1] + \mathbf{v}[k].$$

 $k := k+1.$ **until**  $|f_\delta(\theta[k-1]) - f_\delta(\theta[k-2])| \leq \text{EPS}$  **or**  $k \geq \text{MAXITER}.$ **return**  $\theta = \theta[k-1].$ 

---

される：

$$(P1) \quad \min_{\theta} \|\theta\|_1 \text{ subject to } \|\Phi\theta - \alpha\|_2 \leq \epsilon.$$

この最適化問題は、内点法など標準的な解法を用いても解くことができるが、より高速な解法が知られており、特に Nesterov の方法を用いたアルゴリズムである NESTA (Nesterov's Algorithm) [2] はよく知られた方法である。NESTA の特徴は、まず、(P1) の  $\ell^1$  ノルムの項  $\|\theta\|_1$  を次の微分可能な関数  $f_\delta$  で近似することである (これを Huber 関数と呼ぶ)：

$$f_\delta(\theta) = \sum_{i=1}^N |\theta_i|_\delta, \quad |\theta_i|_\delta := \begin{cases} \frac{1}{2\mu}\theta_i^2, & \text{if } |\theta_i| < \mu, \\ |\theta_i| - \frac{\mu}{2}, & \text{otherwise.} \end{cases}$$

すなわち、問題 (P1) は次の問題として近似される：

$$(P1)_\delta \quad \min_{\theta} f_\delta(\theta) \text{ subject to } \|\Phi\theta - \alpha\|_2 \leq \epsilon.$$

なお、任意の  $\theta \in \mathbb{R}^N$  に対して、 $\delta \rightarrow 0$  のとき  $f_\delta(\theta) \rightarrow \|\theta\|_1$  となる。この近似問題 (P1) $_\delta$  に対して、Nesterov の方法を用いれば、**Algorithm2** に示す NESTA のアルゴリズムが得られる。ここで、 $\text{sat}_\delta(\theta)$  は下記で定義される飽和関数である：

$$[\text{sat}_\delta(\theta)]_i := \begin{cases} \frac{1}{\mu}\theta_i, & \text{if } |\theta_i| < \mu, \\ \text{sgn}(\theta_i), & \text{otherwise.} \end{cases}$$

ただし、 $[\cdot]_i$  はベクトルの第  $i$  要素を示す。なお、 $\text{sat}_\delta(\theta)$  は  $f_\delta(\theta)$  の勾配、すなわち  $\text{sat}_\delta(\theta) = \nabla f_\delta(\theta)$  である。NESTA のアルゴリズムは、任意の初期値に対して、 $O(1/k^2)$  の速さで (P1) $_\delta$  の最適解に収束することが知られている。

次に、問題 (P1) の Lagrange 双対をとった次の最適化問題を考えることもできる。

$$(Q1) \quad \min_{\theta} F(\theta) = \min_{\theta} \|\Phi\theta - \alpha\|_2^2 + \mu\|\theta\|_1.$$

**Algorithm 3** FISTA for sparse control vector  $\theta$ **Require:**  $\alpha \in \mathbb{R}^K$  {observed vector}**Ensure:**  $\theta$  {sparse control vector} $\theta[0] := \mathbf{0}, \tilde{\theta}[1] := \mathbf{0}, \beta[1] := 0.$  $k := 1.$ **repeat** $\theta[k] := \text{shrink}_\mu(\frac{1}{c}\Phi^\top(\alpha - \Phi\tilde{\theta}[k]) + \tilde{\theta}[k]).$  $\beta[k+1] := \frac{1}{2} + \sqrt{\frac{1}{4} + \beta[k]^2}.$  $\tilde{\theta}[k+1] := \theta[k] + \frac{\beta[k]-1}{\beta[k+1]}(\theta[k] - \theta[k-1]).$  $k := k + 1.$ **until**  $|F(\theta[k-1]) - F(\theta[k-2])| \leq \text{EPS}$  or  $k \geq \text{MAXITER}.$ **return**  $\theta = \theta[k-1].$ 

この問題に対しても、高速繰り返し縮小法 (Fast Iterative Shrinkage-Thresholding Algorithm, FISTA) [1] と呼ばれる高速手法が提案され、これにより高速な数値計算が可能となる。FISTA のアルゴリズムを **Algorithm3** に示す。ここで、 $\text{shrink}_\mu$  は

$$[\text{shrink}_\mu(\theta)]_i := \text{sgn}(\eta_i)(|\eta_i| - \lambda/c)_+$$

で定義される非線形関数である。条件  $c \geq \|\Phi\|^2$  が成り立つような  $c > 0$  を選べば、上のアルゴリズムは任意の  $\Phi$  と  $\alpha$  に対して収束し、 $\lim_{k \rightarrow \infty} \theta[k]$  は (Q1) を最小化する最適解に一致する [1]。さらに、最適解への収束の速さは  $O(1/k^2)$  であることも知られている [1]。

本節で述べたアルゴリズムは、計算機への実装が容易であり、また、収束も早いことから、リアルタイムでスパースなベクトル  $\theta$  を得ることが可能である。したがって、リアルタイム性が要求される遠隔制御系に向けたアルゴリズムであると言える。次節では、具体的な遠隔制御系のシミュレーションを用いて、これらのアルゴリズムの優劣を調べる。

**4. シミュレーションによるアルゴリズムの比較**

ここでは、遠隔制御系のシミュレーションにより、前節で紹介したアルゴリズムの優劣を調べる。まず、制御対象  $P$  の伝達関数を

$$\hat{P}(s) = \frac{s - 0.5}{(s + 0.5)(s + 1)}$$

とする。周期  $T$  は  $2\pi$  とする。Fourier 基底  $\{\psi_m\}$  の数は  $N = 2M + 1 = 51$  ( $M = 25$ ) とし、参照信号を

$$r(t) = \sin(5t) + \cos(12.5t)$$

とする。

以上のもと、初期値  $\mathbf{x}_0$  と行列  $U$  をランダムに発生させ、100 回のシミュレーションを行なった。最適化問題 (P0) の解法として、OMP, CoSaMP, NESTA, FISTA の 4 種類を採用し、それらの結果 (推定誤差  $\|\Phi\theta - \alpha\|_2$ , スパース性  $\|\theta\|_0$ , 計算時間 (秒), アルゴリズムの繰り返し回数) の平均値を表 1 に示す。これより、一番性能の良いのは OMP, 一番スパースなベクトルが得られるのは FISTA, 計算時間が最も短いのが FISTA, 繰り返しの回数が一番少ないのが OMP という結果が得られた。

表 1 最適化法の比較

| method | $\ \Phi\theta - \alpha\ _2$ | $\ \theta\ _0$ | CPU time (sec) | # of iteration |
|--------|-----------------------------|----------------|----------------|----------------|
| OMP    | 0.33068                     | 6.66           | 0.026208       | 9.81           |
| CoSaMP | 1.454                       | 5              | 0.056219       | 345.58         |
| NESTA  | 0.62325                     | 51             | 1.183          | 1000           |
| FISTA  | 1.1084                      | 3.79           | 0.0016118      | 17.59          |

なお、NESTA によって得られるベクトル  $\theta$  がスパースでないのは、 $\ell^1$  ノルムを  $f_\delta$  で近似したからであり、0 に非常に近い値は多い。しかし、それを得るには、打ち切りの操作が必要になる。また、FISTA は非常に早く、スパースな解も得られるが、問題は、(Q1) の  $\mu$  の値をどう決めるかである。もし、 $\mu$  の値が適切に定まるのなら、FISTA が遠隔制御系では最も良い選択であるが、もし  $\mu$  の値が不明ならば、OMP が一番良い選択となる。

**5. おわりに**

本稿では、遠隔制御系において、圧縮センシングの手法を用いた場合の、最適化アルゴリズムの優劣をシミュレーションにより調べた。リアルタイム性が必要な遠隔制御系では、FISTA もしくは OMP が適していると考えられる。しかし、最適化アルゴリズムは、他にも様々なものが提案されており、それらを確かめることも重要である。

**文 献**

- [1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sciences*, Vol. 2, No. 1, pp. 183–202 (2009)
- [2] S. Becker, J. Bobin, and E. Candes, NESTA: A Fast and Accurate First-order Method for Sparse Recovery, *SIAM J. Imaging Sci.*, Vol. 4, No. 1, pp. 1–39 (2011)
- [3] E. J. Cnades and M. B. Wakin, An introduction to compressive sampling, *IEEE Signal Processing Magazine*, Vol. 25, pp. 21–30 (2008)
- [4] S. G. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries, *IEEE Trans. Signal Processing*, Vol. 41, No. 12, pp. 3397–3415 (1993)
- [5] 永原, 松田, 林, 圧縮センシングの遠隔制御系への応用, 第 25 回信号処理シンポジウム (2010).
- [6] M. Nagahara and T. Matsuda and K. Hayashi Compressive sampling for remote control systems, *IEICE Trans. on Fundamentals*, Vol. E95-A, No. 4, pp. 713–722 (2012)
- [7] D. Needell and J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, *Applied and Computational Harmonic Analysis*, Vol. 26, No. 3, pp. 301–321 (2009)
- [8] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, *Proc. the 27th Annual Asilomar Conf. on Signals, Systems and Computers*, pp. 40–44 (1993)
- [9] 田中, 圧縮センシングの数理, *IEICE Fundamental Review*, Vol. 4, No. 1, pp. 39–47 (2010)
- [10] J. A. Tropp and S. J. Wright, Computational methods for sparse solution of linear inverse problems, *Proceedings of the IEEE*, Vol. 98, No. 6, pp. 948–958 (2010)
- [11] M. Zibulevsky and M. Elad, L1-L2 optimization in signal and image processing, *IEEE Signal Processing Magazine*, Vol. 27, pp. 76–88 (2010)